



Introduction to fMRIPrep

Lennart Wittkuhn

Max Planck Institute for Human Development

Berlin, Germany

wittkuhn@mpib-berlin.mpg.de

Preprocessing your fMRI data can be like ...



"[...] analysis methods were highly flexible across studies, with nearly as many unique analysis pipelines as there were studies in the sample" (Carp, 2012, *NeuroImage*)

- Overview
 - About: What is fMRIPrep?
 - Workflow: What is fMRIPrep doing?
 - Tools: Which tools is fMRIPrep using?
 - Detour: Brain Imaging Data Structure (BIDS)
- Running fMRIPrep
 - Installation: Docker, Singularity
 - Caveats
 - Running fMRIPrep
 - fMRIPrep outputs and next steps
- Questions, Discussion, Feedback

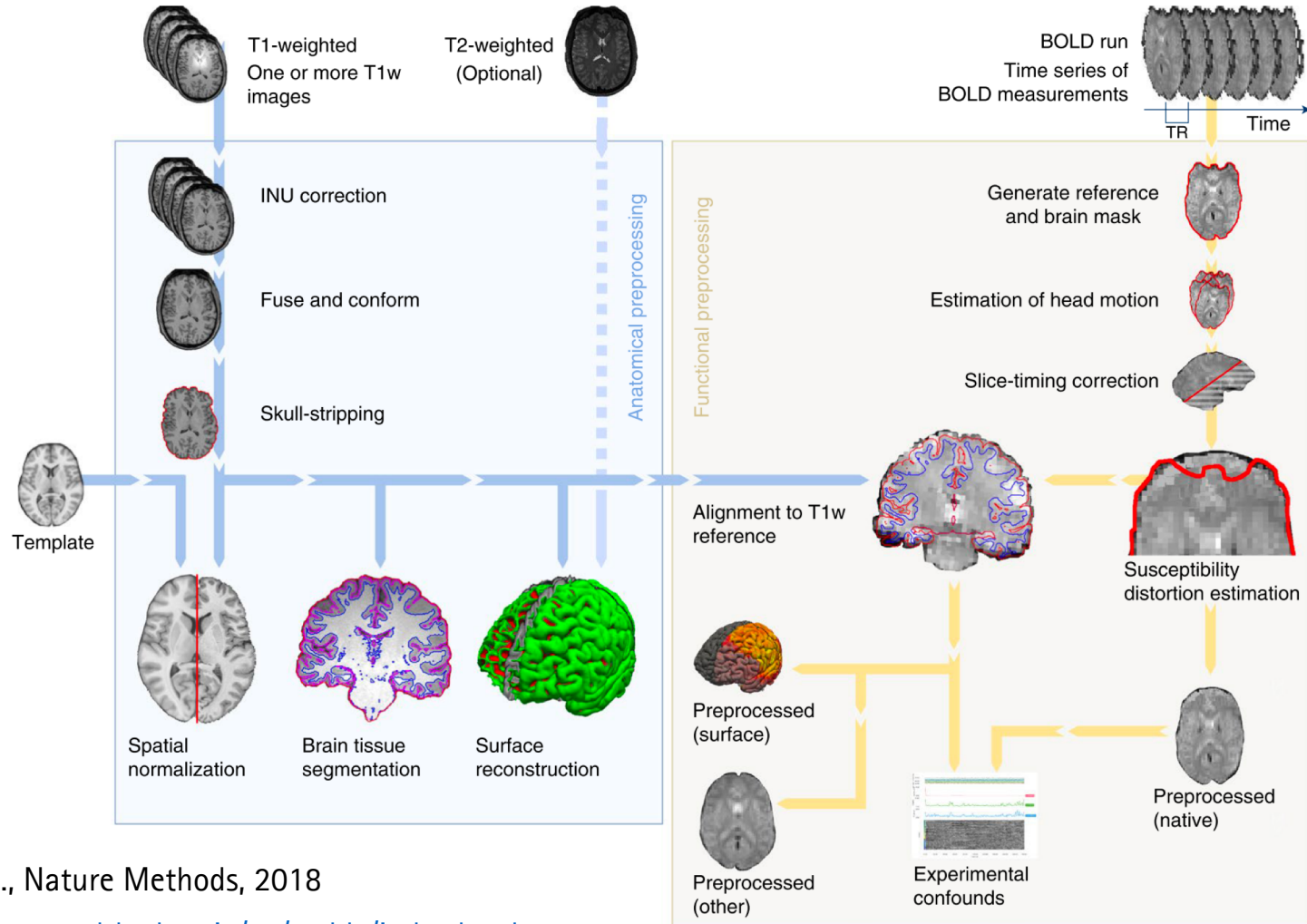
Disclaimer: I am neither an MRI expert nor software developer
Feel free to ask any question throughout the talk!

- a "robust preprocessing pipeline for fMRI data"
- developed by the Poldrack lab (open source with different contributors)
- „provides an **easily accessible**, state-of-the-art interface that is robust to variations in scan acquisition protocols and that **requires minimal user input**, while providing **easily interpretable and comprehensive error and output reporting**"

fMRIPrep principles:

- **Robustness**: pipeline adapts to the input dataset
- **Easy of use**: manual parameter input is reduced to a minimum
- **"Glass box"**: provides visual reports for each subject

fMRIPrep workflow



Esteban et al., Nature Methods, 2018

<https://fmriprep.readthedocs.io/en/stable/index.html>

- Workflow build in Nipype (<https://nipype.readthedocs.io/en/latest/>)
- Flexible combination of different software packages (see below):

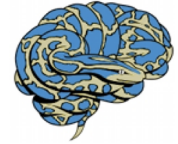


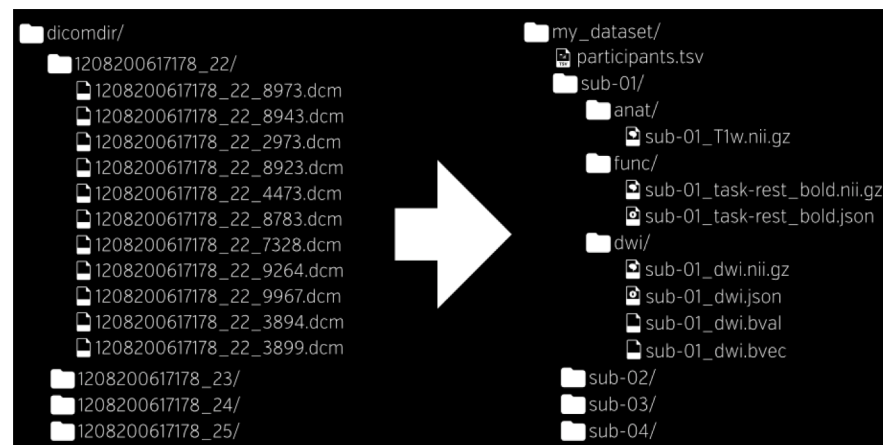
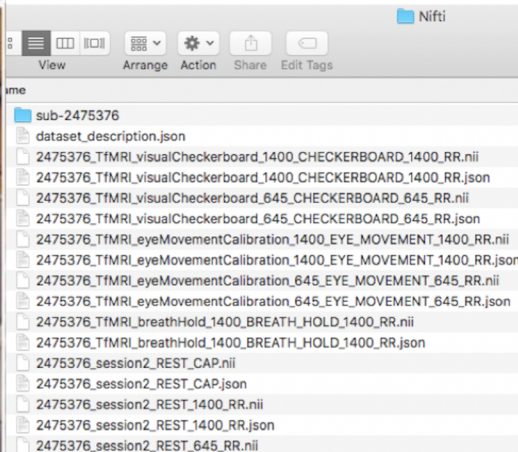
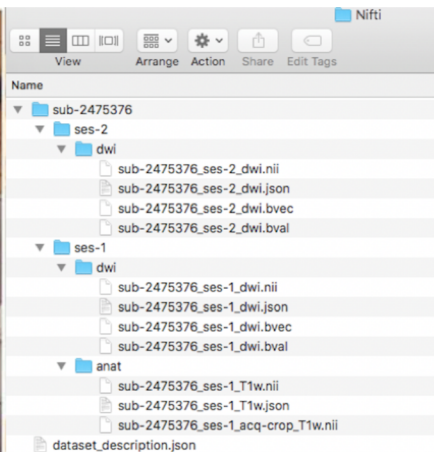
Table 1 | State-of-the-art neuroimaging offers a catalog of readily available software tools

Preprocessing task	Included with fMRIPrep	Alternatives (not included in fMRIPrep)
Anatomical T1-weighted brain extraction	antsBrainExtraction.sh (ANTs)	bet (FSL), 3dSkullstrip (AFNI), MRTOOL (SPM plug-in)
Anatomical surface reconstruction	recon-all (FreeSurfer)	CIVET, BrainSuite, Computational Anatomy (SPM plug-in)
Head-motion estimation (and correction)	MCFLIRT (FSL)	3dvolreg (AFNI), spm_realign (SPM), cross_realign_4dfp (4dfp), antsBrainRegistration (ANTs)
Susceptibility-derived distortion estimation (and unwarping)	3dqwarp (AFNI)	FUGUE and topup (FSL); FieldMap and HySCO (SPM plug-ins)
Slice-timing correction	3dTshift (AFNI)	slicetimer (FSL), spm_slice_timing (SPM), interp_4dfp (4dfp)
Intrasubject registration	bbregister (FreeSurfer), FLIRT (FSL)	3dvolreg (AFNI), antsRegistration (ANTs), Coregister (SPM GUI)
Spatial normalization (intersubject co-registration)	antsRegistration (ANTs)	@auto_tlrc (AFNI), FNIRT (FSL), Normalize (SPM GUI)
Surface sampling	mri_vol2surf (FreeSurfer)	SUMA (AFNI), MNE, Nilearn
Subspace projection denoising (e.g., independent or principal component analysis)	MELODIC (FSL), ICA-AROMA	Nilearn, LMGS (SPM plug-in)
Confounds	In-house implementation	fsl_motion_outliers (FSL), TAPAS PhysIO (SPM plug-in)
Detection of non-steady states	In-house implementation	Ad hoc implementations, manual setting

fMRIPrep integrates best-in-breed tools for each of the preprocessing tasks that its workflow covers, except for steps implemented as part of the development of fMRIPrep (in-house implementations). Tasks listed in the first column are described in detail in Supplementary Note 1.

Requirement: A BIDS-valid dataset

- BIDS aka. the Marie Kondo approach to fMRI datasets

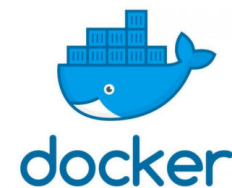


- <https://bids.neuroimaging.io/>
- <https://neurostars.org/t/convert-data-to-bids-format/720>
- <https://bids-standard.github.io/bids-validator/>

1. Upload your dataset to OpenNeuro.org and run fMRIPrep there (or through <https://brainlife.io/> in the future)
2. Run fMRIPrep through a Docker container
 - Problem: Not allowed on most HPCs (high performance clusters)
3. **Solution: Run fMRIPrep in a Singularity container**
 - Build a Singularity image directly by executing this command:

```
singularity build fmriprep-<version>.sim  
docker://poldracklab/fmriprep
```
 - where <version> is the fMRIPrep version you want to use, e.g., latest (see <https://hub.docker.com/r/poldracklab/fmriprep/tags/> for versions):

```
singularity build fmriprep-latest.sim  
docker://poldracklab/fmriprep
```



- fMRIPrep uses FreeSurfer tools, which require a license to run
- To get the license:
 1. Dig up the license in your existing FreeSurfer installation
 2. Register for free here: <https://surfer.nmr.mgh.harvard.edu/registration.html>
- When running fMRIPrep, you will need to either
 1. Specify the path to the license file in the fMRIPrep command (see below)
 2. Set the `$FC_LICENSE` environment variable (see e.g., here: <https://neurostars.org/t/freesurfer-license-errors-with-singularity/1266>)



- Generally, there are different ways to correct for field inhomogeneities, e.g.,:
 - Phase encoding polarity techniques (aka. Blip-up blip-down)
 - Phase difference B0 estimation (e.g., using Siemens GRE sequences)

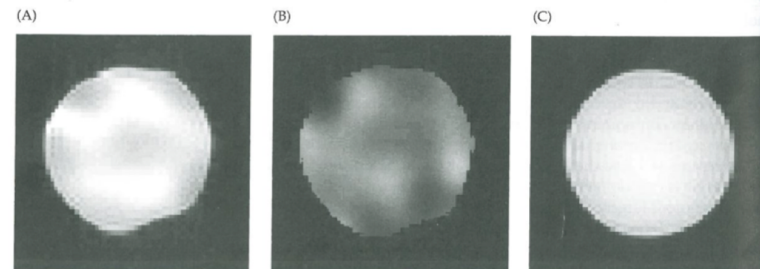


Figure 8.22 Correction of geometric distortions in functional images. (A) In this image of a spherical phantom, there is obvious geometric distortion. (B) To correct for the distortion, a map of magnetic field intensity is acquired. (C) The intensity map can be used to generate a corrected image.

- The .json file of the fieldmaps need to point to the fMRI files by adding a IntendedFor field to the .json file:

```
},  
"IntendedFor": [  
  "ses-01/func/sub-01_ses-01_task-highspeed_rec-prenorm_run-01_bold.nii.gz",  
  "ses-01/func/sub-01_ses-01_task-highspeed_rec-prenorm_run-02_bold.nii.gz",  
  "ses-01/func/sub-01_ses-01_task-highspeed_rec-prenorm_run-03_bold.nii.gz",  
  "ses-01/func/sub-01_ses-01_task-highspeed_rec-prenorm_run-04_bold.nii.gz",  
  "ses-01/func/sub-01_ses-01_task-rest_rec-prenorm_run-01_bold.nii.gz"  
]
```

- Ideally, fMRIPrep will automatically find and apply the fieldmap files
- Either add the IntendedFor field manually or (better) write a custom script

```
lip-osx-003854:Desktop wittkuhn$ singularity run --cleanenv  
${PATH_TO_SINGULARITY_IMAGE} --fs-license-file  
${PATH_TO_FREESURFER_LICENSE} ${PATH_TO_BIDS_DIRECTORY}  
${PATH_TO_OUTPUT_DIRECTORY} participant --participant_label  
${SUBJECT_ID} -w ${PATH_TO_WORKING_DIRECTORY} --mem_mb  
${MEM_MB} --nthreads ${N_CPUS} --omp-nthreads ${N_THREADS} --  
write-graph --stop-on-first-crash --output-space T1w fsnative  
template fsaverage --notrack --verbose --resource-monitor
```

```
${PATH_TO_SINGULARITY_IMAGE} → ../fmriprep-1.1.8.simg  
${PATH_TO_FREESURFER_LICENSE} → ../freesurfer_license.txt  
${PATH_TO_BIDS_DIRECTORY} → awesome_project/bidsdata/  
${PATH_TO_OUTPUT_DIRECTORY} → awesome_project/derivatives/  
${SUBJECT_ID} → sub-01  
${PATH_TO_WORKING_DIRECTORY} → awesome_project/work/  
${MEM_MB} → 30000  
${N_CPUS} → 8  
${N_THREADS} → 8
```

fMRIPrep generates three broad classes of outcomes:

- **Visual QA (quality assessment) reports:** one HTML per subject, that allows the user a thorough visual assessment of the quality of processing and ensures the transparency of fMRIPrep operation.
- **Pre-processed imaging data** which are derivatives of the original anatomical and functional images after various preparation procedures have been applied
- **Additional data for subsequent analysis**, for instance the transformations between different spaces or the estimated confounds.

Next steps:

- Remember fMRIPRep is performing **minimal** preprocessing!
- But, additional steps (smoothing, GLM, etc.) are easy to hook up

Some (personal) praise for fMRIPrep

- Works (almost) out-of-the-box
- You will (most likely) not code up a pipeline this accurate yourself

Limitations and reasons not to use fmriprep (from fMRIPrep website)

- Potential image registration problems with narrow FoV images
- fMRIPrep may underperform for particular populations (e.g., infants)
- You really want unlimited flexibility (obviously a double-edged sword).
- If you are trying to reproduce some *in-house* lab pipeline.
- If you want students to suffer through implementing each step for didactic purposes, or to learn shell-scripting or Python along the way.

- Danger of „blind trust“ and lack of understanding of the preprocessing steps
- Potential inflexibility for specific use-cases

- Documentation: <https://fmriprep.readthedocs.io/en/stable/index.html>
- Paper: Esteban et al., 2018, Nature Methods
- Slides by Chris Gorgolewski: https://de.slideshare.net/chrisfilo1/fmriprep-robust-and-easy-to-use-fmri-preprocessing-pipeline?next_slideshow=1
- <https://neurostars.org/> for questions



fMRIPrep: a robust preprocessing pipeline for functional MRI

Oscar Esteban^{1*}, Christopher J. Markiewicz², Ross W. Blair¹, Craig A. Moodie³, A. Ilkay Isik², Asier Erramuzpe³, James D. Kent⁴, Mathias Goncalves⁵, Elizabeth DuPre⁶, Madeleine Snyder⁷, Hiroyuki Oya⁸, Satrajit S. Ghosh^{5,9}, Jesse Wright¹, Joke Durnez¹, Russell A. Poldrack^{1,10} and Krzysztof J. Gorgolewski^{1,10*}

- My slides will be distributed by Matthias
- Feel free to email me: wittkuhn@mpib-berlin.mpg.de

Confounds from fmriprep: which one would you use for GLM?

fmriprep



mvdoc

Mar '17

Hello experts,

after running fmriprep, I get a very nice tsv with the computed confounds. Would you use **all of them** as nuisance regressors in a GLM? Or are some more suited for resting state data vs. task-based data?

Thanks!

Matteo

3

Conducting First Level GLM following fmriprep ²

created

Mar '17

last reply

Feb 27

32

replies

4.6k

views

15

users

22

likes

11

links



ChrisGorgolewski Regular

Mar '17

Opinions on this topic are divided and this is why FMRIPREP provides those regressors instead of cleaning up the data for you. I personally would include 6 motion parameters, FD, and aCompCor on run level and mean FD on group level (for both task and rest).

- <https://neurostars.org/t/initial-volumes-to-delete-before-first-level-analysis-with-a-multiband-fmri-data/461>



ChrisGorgolewski  Regular

May '17

- SPM8 and 12 should support 4D files
- Most modern scanners discard non-steady state (dummy) scans these days. We have an automatic detector of dummy scans built into FMRIPREP, but we are not exposing this information. You can just look at your data and check if the first few volumes are brighter, or alternatively run MRIQC which also has a built in detector and will indicate in the report if dummy scans were detected.

1  